

(19)日本国特許庁(J P)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平5-81219

(49)公報日 平成5年(1993)4月2日

(51)IntCl<sup>4</sup>G 0 6 F 15/18  
15/78

識別記号

4 2 0 S 9190-5L  
5 1 0 A 7530-5L

片内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数1(全 13 頁)

(21)出願番号 特願平3-273228

(22)出願日 平成3年(1991)9月24日

(71)出願人 000004260

日本電装株式会社

愛知県刈谷市昭和町1丁目1番地

(72)発明者 鈴木 隆夫

愛知県刈谷市昭和町1丁目1番地 日本電  
装株式会社内

(72)発明者 神谷 敏玄

愛知県刈谷市昭和町1丁目1番地 日本電  
装株式会社内

(72)発明者 川崎 孝二

愛知県刈谷市昭和町1丁目1番地 日本電  
装株式会社内

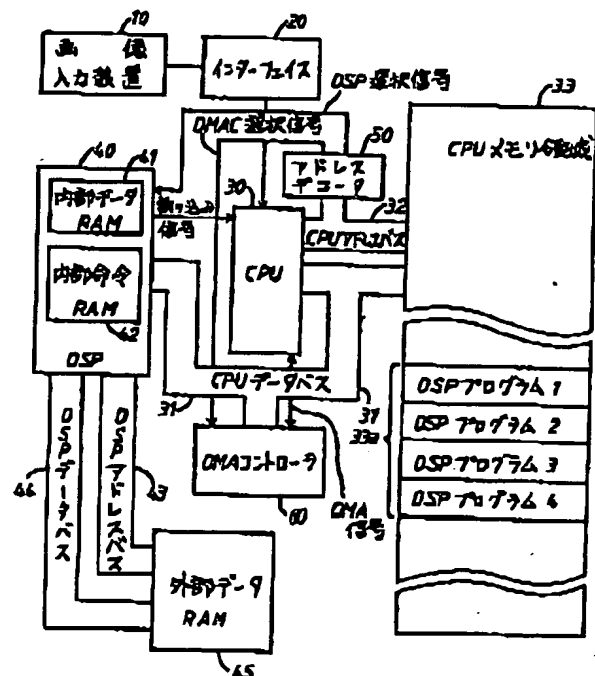
(74)代理人 弁理士 長谷 照一

(54)【発明の名称】 マイクロプロセッサ動作制御装置

(57)【要約】

【目的】 ホストマイクロプロセッサからDSPの内部命令RAMへの命令コードの転送によるDSPの実行速度の低下を最小限に抑えるために、前回と同じ処理を行う場合には命令コードの転送を行わないようにする。

【構成】 CPU30が、DSP40の各処理内容にそれぞれ対応する各処理番号を内部データRAM41に選択的にセットし、このセット毎にDSP40の実行をスタートさせる。このスタート毎に、内部データRAM41内のセット処理番号が内部命令RAM42内のDSPプログラムの番号に対応するとき、DSP40が、内部命令RAM42内のDSPプログラムに基づきこれに対応する処理内容を実行する。DSP40による対応しないとの判別するとき、ホストマイクロプロセッサが、内部データRAM41にセット済みの処理番号に対応するプログラム番号のDSPプログラムを内部命令RAM42に転送記憶させる。



1

## 【特許請求の範囲】

【請求項1】ホストマイクロプロセッサと、内部データRAM及び内部命令RAMを有しスレーブマイクロプロセッサとしての役割を果たすDSPとを備え、前記ホストマイクロプロセッサが、前記DSPの複数の処理内容をそれぞれ表す複数のDSPプログラムをその各プログラム番号と共に予め記憶する記憶手段と、

前記DSPの複数の処理内容にそれぞれ対応する各処理番号を前記内部データRAMに選択的にセットする処理番号セット手段と、

この処理番号セット手段によるセット後前記DSPの実行をスタートさせるスタート手段とを有し、

また、前記DSPが、

同DSPのスタートのもとに前記内部データRAM内のセット処理番号が前記内部命令RAM内のDSPプログラムの番号に対応するか否かにつき判断する判断手段と、

この判断手段の対応するとの判断のとき前記内部命令RAM内のDSPプログラムに基づきこれに対応する処理内容を実行する実行手段とを有し、

また、前記判断手段の対応しないとの判断のとき、前記ホストマイクロプロセッサが、前記内部データRAMにセット済みの処理番号に対応するプログラム番号のDSPプログラムを前記記憶手段から前記内部命令RAMに転送するようにしたマイクロプロセッサ動作制御装置。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】本発明は、チップ内部に命令ランダム・アクセス・メモリ（以下、内部命令RAMという）を有し、この内部命令RAMから読み出した命令に従って動作を実行するマイクロプロセッサに係り、特に、当該マイクロプロセッサのうちのデジタルシグナルプロセッサ（以下、DSPという）の命令実行を制御するに適したマイクロプロセッサ動作制御装置に関する。

## 【0002】

【従来技術】従来、内部命令RAMを有しスレーブマイクロプロセッサとして機能するDSPにおいては、実行に先立ち、ホストマイクロプロセッサのセントラル・プロセッシング・ユニット（以下、CPUという）がその内蔵メモリから命令コードを読み出しDSPの内部命令RAMに転送するようにしてあるのが通常である。

## 【0003】

【発明が解決しようとする課題】ところで、このようなDSPではチップ面積の制約から内部命令RAMの容量が数百ステップに制限されるのが普通である。従って、多くの処理内容を単一のDSPで行う場合には内部命令RAM内の命令コード内容を入れ換えることが必要となるという不具合がある。しかし、DSPの各処理内容の

(2)

特開平5-81219

2

実行に先だって、毎回、内部命令RAM内の命令コード内容を入れ換えることはDSPの実行速度の低下を招くという不具合がある。また、開発段階でDSPの命令コード或いはプログラム構成を変更すると、CPUもこの変更を把握しないと命令コードの転送ができないため、DSPのプログラム変更に従ってCPUのプログラム変更が必要となるという不具合を生ずる。そこで、本発明は、以上のようなことに対処すべく、マイクロプロセッサ動作制御装置において、ホストマイクロプロセッサからDSPの内部命令RAMへの命令コードの転送によるDSPの実行速度の低下を最小限に抑えるために、前回と同じ処理を行う場合には命令コードの転送を行わないようにするとともに、DSPのプログラムを変更してもホストマイクロプロセッサのプログラムの変更を不要にするようにしようとするものである。

## 【0004】

【課題を解決するための手段】上記課題の解決にあたり、本発明の構成は、図1にて例示することく、ホストマイクロプロセッサと、内部データRAM及び内部命令RAMを有しスレーブマイクロプロセッサとしての役割を果たすDSPとを備え、前記ホストマイクロプロセッサが、前記DSPの複数の処理内容をそれぞれ表す複数のDSPプログラムをその各プログラム番号と共に予め記憶する記憶手段1と、前記DSPの複数の処理内容にそれぞれ対応する各処理番号を前記内部データRAMに選択的にセットする処理番号セット手段2と、この処理番号セット手段2によるセット後前記DSPの実行をスタートさせるスタート手段3とを有し、また、前記DSPが、同DSPのスタートのもとに前記内部データRAM内のセット処理番号が前記内部命令RAM内のDSPプログラムの番号に対応するか否かにつき判断する判断手段4と、この判断手段4の対応するとの判断のとき前記内部命令RAM内のDSPプログラムに基づきこれに対応する処理内容を実行する実行手段5とを有し、また、判断手段4の対応しないとの判断のとき、前記ホストマイクロプロセッサが、前記内部データRAMにセット済みの処理番号に対応するプログラム番号のDSPプログラムを記憶手段1から前記内部命令RAMに転送するようにしたことにある。

## 【0005】

【作用】前記ホストマイクロプロセッサの処理番号セット手段2が、前記DSPの複数の処理内容にそれぞれ対応する各処理番号を前記内部データRAMに選択的にセットすると、このセット毎にスタート手段3が前記DSPの実行をスタートさせる。すると、このスタート毎に、前記DSPの判断手段4が、前記内部データRAM内のセット処理番号が前記内部命令RAM内のDSPプログラムの番号に対応するか否かにつき判断し、実行手段5が、判断手段4の対応するとの判断のとき、前記内部命令RAM内のDSPプログラムに基づきこれに対応

(3)

特開平5-81219

3

する処理内容を実行する。また、判断手段4の対応しないとの判断のとき、前記ホストマイクロプロセッサが、前記内部データRAMにセット済みの処理番号に対応するプログラム番号のDSPプログラムを記憶手段1から前記内部命令RAMに転送記憶させる。すると、上述と同様の判断手段4による対応するとの判断のもとに、実行手段5が、前記内部命令RAMに転送記憶させたDSPプログラムに基づきこれに対応する処理内容を実行する。

【0008】

【発明の効果】このように、判断手段4の対応しないとの判断がない限り、前記内部命令RAM内のDSPプログラムを入れ換えることなくそのまま使用して実行手段5の実行が行われるので、前記内部命令RAM内のDSPプログラムの入れ換えによるDSPの実行速度の低下を、前記内部命令RAMの容量を増大させることなく、最小限に抑制し得る。また、前記内部命令RAM内のDSPプログラムの入れ換えにあたっては、判断手段4の対応しないとの判断に基づき前記ホストマイクロプロセッサが記憶手段1から前記内部命令RAMへDSPプログラムを転送するのみでよいので、前記ホストマイクロプロセッサのプログラムの変更は不要である。

【0007】

【実施例】以下、本発明の一実施例を図面により説明すると、図2は、本発明が指紋照合システムに適用された例を示している。この指紋照合システムは、画像入力装置10を備えており、この画像入力装置10は、そのイメージセンサの読みとり画面に指紋照合者の指が押し付けられたとき、同指紋照合者の指紋を光学的に読み取り、この読み取り指紋画像を指紋画像出力信号として発生する。インターフェイス20は画像入力装置10からの指紋画像出力信号をホストマイクロプロセッサのCPU\*

4

\*U30に出力する。CPU30は、インターフェイス20、DSP40及びDMAコントローラ60等との協働により、図3～図6に示すフローチャートに従い主制御プログラム及び割り込み制御プログラムを実行し、この実行中において、指紋照合に必要な演算処理をする。また、CPU30は、メモリ（以下、CPUメモリという）を内蔵しており、このCPUメモリのメモリ領域33（以下、CPUメモリ領域33という）のDSP40に対するDSP命令コード保持領域33aには、4個のDSPプログラム1、2、3及び4が予め記憶されている。なお、上述の主制御プログラム及び割り込み制御プログラムは、CPUメモリ領域33内に予め記憶されている。

10

【0008】DSP40は、スレーブマイクロプロセッサとしての役割をもち、コミュニケーション領域を有するデータRAM41（以下、内部データRAMという）及び内部命令RAM42を内蔵しており、内部命令RAM42はDSP40の命令内容をデータとして一時的に記憶する。また、DSP40は、図7にて示すフローチャートに従いDSP制御プログラムを実行し、この実行中において、種々の演算処理をしてその各終了によりCPU30に割り込みをかけるとき割り込み信号を発生し同CPU30に出力する。また、DSP40は、DSPアドレスバス43及びDSPデータバス44を介し外部データRAM45へ同外部データRAM45に一時的に記憶させるべくデータを出力する。なお、上述のDSP制御プログラムはDSP40内に予め記憶されている。

【0009】また、DSP40で行う各処理内容の各処理番号と各プログラム番号は、DSP40の各処理内容との関係において、次の表1の通り対応する。

【表1】

処理番号	処理内容	プログラム番号
1	マニユーシャの位置及び方向の計算	1
2	疑似マニユーシャ除去	2
3	位置ずれ及び回転量の計算	2
4	アフィン変換	3

但し、各処理番号1～4は、表1のごとく、DSP40の各処理内容に対し1対1でもって対応する一方、DSP40の各処理内容と各プログラム番号との対応関係は、マニユーシャの位置及び方向の計算と位置ずれ及び回転量の計算との両処理内容に対し単一のプログラム番

号2が対応している。なお、各プログラム番号はDSP命令コード保持領域33aの何番のプログラムにDSP40の各処理内容が入っているかを表す。

【0010】次の表2は、プログラム番号テーブルを表しており、このプログラム番号テーブルにおいては、同

50

(4)

特開平5-81219

5

テーブルの上欄から下欄に向けて表1のプログラム番号が処理番号順に入っている。但し、このプログラム番号テーブルは、DSP40の内部データRAM41に記憶\*

\*されている。  
【表2】

6

オフセットアドレス	テーブル内容
1	1
2	2
3	2
4	3

【0011】次の表3は、DSP40のコミュニケーション領域にセットされる各データを表すもので、これら各データは、DSP40とCPU30との間で受け渡される。

処理番号
終了フラグ
プログラム変更要求フラグ
プログラム番号

【0012】アドレスデコード50は、CPU30がDSP40にアクセスする必要があるとき同CPU30からのアクセス指令をデコードしDSP選択信号としてDSP40に出力する。また、このアドレスデコード50は、CPU30がダイレクト・メモリ・アクセス・コントローラ50（以下、DMAコントローラ60という）にアクセスするとき同CPU30からのアクセス指令をデコードしDMA選択信号としてDMAコントローラ60に出力する。DMAコントローラ60は、アドレスデコード50からのDMA選択信号及びCPU30からのDMA信号にตอบสนองしてCPU30からCPUデータバス31を通しDSP40の内部データRAM41及び内部命令RAM42へ必要なデータを転送する。

【0013】以上のように構成した本実施例において、本発明システムを作動状態におけば、CPU30が、図3～図5のフローチャートに従い主制御プログラムの実行をステップ100にて開始する。現段階において、画像入力装置10から指紋画像出力信号がインターフェイス20に出力されておれば、CPU30が、ステップ101にて、同インターフェイス20から同指紋画像出力信号を入力される。すると、CPU30が、ステップ102にて、同指紋画像出力信号の値を二値化して指紋の隆線山画像を作成し、ステップ103にて、同隆線山画

像を反転して指紋の隆線谷画像を作成し、ステップ104にて、ステップ102における隆線山画像を細線化し、かつ、ステップ105にて、DSP40の処理内容のうちマニユーシャの位置及び方向の計算に対応する処理番号1を、DSP40の内部データRAM41のコミュニケーション領域内にセットして主制御プログラムを次のステップ106に進める。

【0014】このようにして主制御プログラムがステップ106に進むと、CPU30が、DSP40をスタートさせ、次のステップ107にて、ステップ103における隆線谷画像を細線化し、ステップ110にて「NO」との判別を繰り返す。また、上述のようにDSP40をスタートさせると、DSP40が、図7のフローチャートに従い、DSP制御プログラムの実行を、ステップ300にて開始する。ついで、DSP40が、次のステップ310にて、内部データRAM41のコミュニケーション領域内にセット済みの処理番号1を読み出してDSP制御プログラムをステップ320に進める。

【0015】しかし、ステップ310にて読み出した処理番号1が、内部命令RAM42に転送済みのDSPプログラムで実行できる処理番号と一致しない場合には、DSP40が、ステップ320にて「NO」と判別し、ステップ350にて、内部データRAM41内のプログラム番号テーブルを参照し、ステップ310にて読み出し済みの処理番号1に対応するプログラム番号1を内部データRAM41のコミュニケーション領域にセットし、かつ、ステップ360にて、プログラム変更要求フラグを前記コミュニケーション領域にセットしてDSP制御プログラムをステップ370に進める。

【0016】このようにしてDSP制御プログラムがステップ370に進むと、DSP40がCPU30に割り込みをかける。すると、CPU30が、主制御プログラムの実行を中止して図6のフローチャートに従い割り込み制御プログラムの実行をステップ200にて開始し、かつステップ210にて、終了フラグのセットの有無をチェックする。現段階において、上述のようにステップ

50

320における「NO」との判別のもとに終了フラグがセットされないため、CPU30が、ステップ210における未終了との判定のもとに、ステップ230にて、プログラム変更要求フラグのセットの有無をチェックする。このとき、上述のようにステップ360にてプログラム変更要求フラグがセットされているため、CPU30が、ステップ230にて、プログラム変更要求ありとの判定をし、ステップ250にて、内部データRAM41のコミュニケーション領域からプログラム番号1を取得する。

【0017】しかし、プログラム番号1で指定されたDSPプログラムをDSP40の内部命令RAM42に転送すべく、CPU30が、ステップ260にて、アドレスデコーダ50からDMAC選択信号をDMAコントローラ60に出力させると、このDMAコントローラ60が、CPU30のCPUメモリ領域33におけるDSP命令コード保持領域33aからDSPプログラム1を読み出して内部命令RAM42に転送して新たに記憶させる。ついで、CPU30が、ステップ270にて、DSP40をスタートさせ、次のステップ280にて、割り込み制御プログラムの実行を終了する。

【0018】上述のようにステップ270にてDSP40をスタートさせると、同DSP40が、図7のフローチャートに従いDSP制御プログラムの実行をステップ300にて開始し、ステップ310にて内部データRAM41のコミュニケーション領域内の処理番号1を読み出す。現段階にては、上述のように読み出した処理番号1が内部命令RAM42にステップ260にて転送済みのDSPプログラムに対応するため、DSP40が、ステップ320にて「YES」と判別し、内部命令RAM42内に転送済みのDSPプログラムに基づき、ステップ330にて、ステップ104における壁線山画像細線化結果に基づき、マニューシャの位置及び方向を計算する処理を実行し、ステップ340にて、終了フラグを内部データRAM41のコミュニケーション領域にセットし、かつ、ステップ370にて、CPU30に割り込みをかける。

【0019】このため、CPU30が、上述と同様に割り込み制御プログラムの実行を開始し、ステップ210にて、ステップ340における終了フラグのセット済みのもとに、終了と判定し、かつ、ステップ220にて割り込み制御プログラムの実行を終了する。然る後、CPU30が、主制御プログラムのステップ110にて、ステップ340における終了フラグのセットに基づき「YES」と判別し、ステップ111にて、疑似マニューシャ除去の処理番号2を内部データRAM41のコミュニケーション領域にセットし、かつ、次のステップ112にてDSP40をスタートさせ、かつ、ステップ120にて「NO」との判別を繰り返す。

【0020】上述のようにステップ112にてDSP4

0をスタートさせると、DSP40が、図7のフローチャートに従い、DSP制御プログラムのステップ310にて、内部データRAM41のコミュニケーション領域内にセット済みの処理番号2を読み出してDSP制御プログラムをステップ320に進める。しかし、現段階にては、ステップ310にて読み出した処理番号2が、内部命令RAM42に転送済みのDSPプログラムで実行できる処理番号1と一致しないため、DSP40が、ステップ320にて「NO」と判別し、ステップ350にて、内部データRAM41内のプログラム番号テーブルを参照し、ステップ310にて読み出し済みの処理番号2に対応するプログラム番号2を内部データRAM41のコミュニケーション領域にセットし、かつ、ステップ360にて、プログラム変更要求フラグを前記コミュニケーション領域にセットしてDSP制御プログラムをステップ370に進めCPU30に割り込みをかける。

【0021】このようにしてCPU30に割り込みをかけると、CPU30が、主制御プログラムの実行を中止して図8のフローチャートに従い割り込み制御プログラムの実行に移行し、ステップ210にて、上述のようなステップ320における「NO」との判別のもとに、未終了と判定し、ステップ230にて、上述のようなステップ360におけるプログラム変更要求フラグのセットに基づき、プログラム変更要求ありとの判定をし、ステップ250にて、内部データRAM41のコミュニケーション領域からプログラム番号2を取得する。しかし、プログラム番号2で指定されたDSPプログラムをDSP40の内部命令RAM42に転送すべく、CPU30が、ステップ260にて、アドレスデコーダ50からDMAC選択信号をDMAコントローラ60に出力させると、このDMAコントローラ60が、CPU30のCPUメモリ領域33におけるDSP命令コード保持領域33aからDSPプログラム2を読み出して内部命令RAM42に転送して新たに記憶させる。ついで、CPU30が、ステップ270にて、DSP40をスタートさせ、次のステップ280にて、割り込み制御プログラムの実行を終了する。

【0022】このようにステップ270にてDSP40をスタートさせると、同DSP40が、図7のフローチャートに従いDSP制御プログラムを実行し、ステップ310にて、内部データRAM41のコミュニケーション領域内の処理番号2を読み出す。現段階にては、上述のように読み出した処理番号2が内部命令RAM42にステップ260にて転送済みのDSPプログラムに対応するため、DSP40が、ステップ320にて「YES」と判別し、内部命令RAM42内に転送済みのDSPプログラムに基づき、ステップ330にて、上述のマニューシャの位置及び方向の計算結果に基づき疑似マニューシャ除去の処理を実行し、ステップ340にて、終了フラグを内部データRAM41のコミュニケーション

9

ン領域にセットし、かつ、ステップ370にて、CPU30に割り込みをかける。このため、CPU30が、上述と同様に割り込み制御プログラムの実行を開始し、ステップ210にて、ステップ340における終了フラグのセット済みのもとに、終了と判定し、かつ、ステップ220にて割り込み制御プログラムの実行を終了する。

【0023】然る後、CPU30が、主制御プログラムのステップ120にて、ステップ340における終了フラグのセットに基づき「YES」と判別し、ステップ121にて、位置ずれ及び回転量計算のための処理番号3を内部データRAM41のコミュニケーション領域にセットし、かつ、次のステップ122にてDSP40をスタートさせ、かつステップ130にて「NO」との判別を繰り返す。このようにステップ122にてDSP40をスタートさせると、DSP40が、図7のフローチャートに従い、DSP制御プログラムのステップ310にて、内部データRAM41のコミュニケーション領域内にセット済みの処理番号3を読み出してDSP制御プログラムをステップ320に進める。

【0024】現段階にては、上述のように読み出した処理番号3が内部命令RAM42にステップ260にて転送済みのDSPプログラムに対応するため、DSP40が、ステップ320にて「YES」と判別し、内部命令RAM42内に転送済みのDSPプログラム2に基づき、ステップ330にて、上述のマニューシャの位置及び方向の計算結果に基づき、位置ずれ及び回転量を計算する処理を実行し、ステップ340にて、終了フラグを内部データRAM41のコミュニケーション領域にセットし、かつ、ステップ370にて、CPU30に割り込みをかける。このため、CPU30が、上述と同様に割り込み制御プログラムの実行を開始し、ステップ210にて、ステップ340における終了フラグのセット済みのもとに、終了と判定し、かつ、ステップ220にて割り込み制御プログラムの実行を終了する。

【0025】然る後、CPU30が、主制御プログラムのステップ130にて、ステップ340における終了フラグのセットに基づき「YES」と判別し、ステップ131にて、アフィン変換の処理番号4を内部データRAM41のコミュニケーション領域にセットし、かつ、次のステップ132にてDSP40をスタートさせ、かつ、ステップ140にて「NO」との判別を繰り返す。このようにステップ132にてDSP40をスタートさせると、DSP40が、図7のフローチャートに従い、DSP制御プログラムのステップ310にて、内部データRAM41のコミュニケーション領域内にセット済みの処理番号4を読み出してDSP制御プログラムをステップ320に進める。

【0026】現段階にては、上述のように読み出した処理番号4が内部命令RAM42にステップ260にて転送済みのDSPプログラムに対応しないため、DSP4

(6)

特開平5-81219

10

0が、ステップ320にて「NO」と判別し、ステップ350にて、内部データRAM41内のプログラム番号テーブルを参照し、ステップ310にて読み出し済みの処理番号4に対応するプログラム番号3を内部データRAM41のコミュニケーション領域にセットし、かつ、ステップ360にて、プログラム変更要求フラグを前記コミュニケーション領域にセットしてDSP制御プログラムをステップ370に進め、CPU30に割り込みをかける。

10 【0027】すると、CPU30が、主制御プログラムの実行を中止して図6のフローチャートに従い割り込み制御プログラムをの実行し、ステップ210にて、上述と同様に未終了と判定し、ステップ230にて、上述のようなステップ360におけるプログラム変更要求フラグのセットに基づき、プログラム変更要求ありとの判定をし、ステップ250にて、内部データRAM41のコミュニケーション領域からプログラム番号4を取得する。しかして、プログラム番号4で指定されたDSPプログラムをDSP40の内部命令RAM42に転送すべく、CPU30が、ステップ260にて、アドレスデコード50からDMAC選択信号をDMAコントローラ80に出力させると、このDMAコントローラ80が、CPU30のCPUメモリ領域33におけるDSP命令コード保持領域33aからDSPプログラム4を読み出して内部命令RAM42に転送して新たに記憶させる。ついで、CPU30が、ステップ270にて、DSP40をスタートさせ、次のステップ280にて、割り込み制御プログラムの実行を終了する。

30 【0028】上述のようにステップ270にてDSP40をスタートさせると、同DSP40が、図7のフローチャートに従いDSP制御プログラムを実行し、ステップ310にて内部データRAM41のコミュニケーション領域内の処理番号4を読み出す。現段階にては、上述のように読み出した処理番号4が内部命令RAM42にステップ260にて転送済みのDSPプログラムに対応するため、DSP40が、ステップ320にて「YES」と判別し、内部命令RAM42内に転送済みのDSPプログラム4に基づき、ステップ330にて、上述の位置ずれ及び回転量計算結果に基づきアフィン変換の処理を実行し、ステップ340にて、終了フラグを内部データRAM41のコミュニケーション領域にセットし、かつ、ステップ370にて、CPU30に割り込みをかける。

40 【0029】このため、CPU30が、上述と同様に割り込み制御プログラムの実行を開始し、ステップ210にて、ステップ340における終了フラグのセット済みのもとに、終了と判定し、かつ、ステップ220にて割り込み制御プログラムの実行を終了する。然る後、CPU30が、ステップ140にて、上述と実質的に同様に「YES」と判別し、ステップ150にて、指紋照合者

50

## II

の登録指紋画像とアフィン変換後の照合指紋画像とのパターンマッチングを行い、ステップ180にて、同パターンマッチング結果に基づき、登録指紋画像と照合指紋画像との一致不一致を判定する。但し、上述の登録指紋画像はCPU30のCPUメモリ領域に予め記憶されている。

【0030】以上説明したように、指紋照合者の指紋照合にあたり、上述のようなCPU30とDSP40との協働による演算処理過程において、疑似マニューシャ除去の処理と位置ずれ及び回転量の計算の処理とが上述のように続くときは、主制御プログラムにおけるステップ111～ステップ122の演算処理並びにDSP制御プログラムの実行を通じ、ステップ330における各疑似マニューシャ除去の処理と位置ずれ及び回転量の計算の処理とを、内部命令RAM42内のDSPプログラム2を変更することなくそのまま維持した上で実行するので、内部命令RAM42へのDSPプログラムの入れ換え回数を最小限に抑制させることができ、その結果、内部命令RAM42の容量を最小限にしつつDSP40の実行速度を向上させ得る。かかる場合、内部命令RAM42内のDSPプログラムの入れ換えはCPU30のプログラムの変更を必要としない。また、主制御プログラムのステップ104における隆線山画像の細線化処理後、ステップ106にてスタートしたDSP40によりステップ330における隆線山画像のマニューシャ位置及び方向計算をしながら、CPU30における隆線谷画像の細線化を並行処理するので、CPU30及びDSP\*

(7)

特開平5-81219

12

\*40の協働による実行処理速度を向上させることができる。

【0031】なお、本発明の実施にあたり、DMPコントローラ60を利用しない場合には、CPU30からのI/Oリード・ライトでもDSP40側へのアクセスを行うことができる。また、本発明の実施にあたっては、指紋照合システムに限ることなく、DSP及びホストCPUを備えた各システムに本発明を適用して実施してもよい。

## 10 【図面の簡単な説明】

【図1】 特許請求の範囲の記載に対する対応図である。

【図2】 本発明を適用した指紋照合システムのブロック図である。

【図3】 図2のCPUにより実行すべき主制御プログラムのフローチャートの前段部である。

【図4】 同主制御プログラムのフローチャートの中段部である。

【図5】 同主制御プログラムのフローチャートの後段部である。

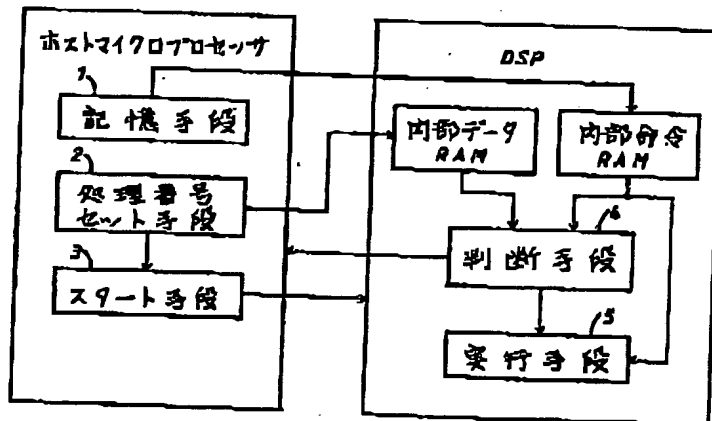
## 20 【図6】 図2のCPUにより実行すべき割り込み制御プログラムのフローチャートである。

【図7】 図2のDSPにより実行すべきDSP制御プログラムのフローチャートである。

【符号の説明】

30…CPU、33…CPUメモリ領域、40…DSP、41…内部データRAM、42…内部命令RAM。

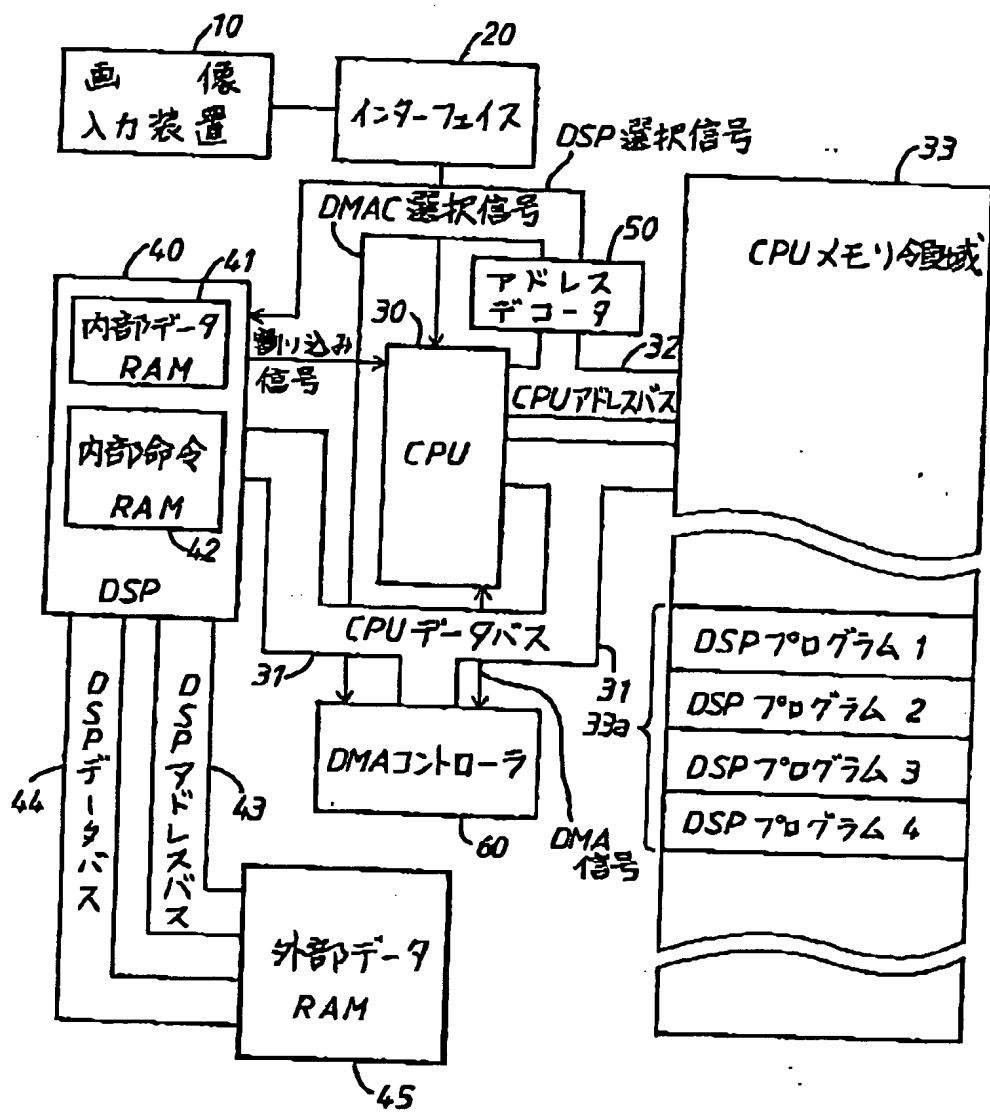
【図1】



(8)

特開平5-81219

【図2】

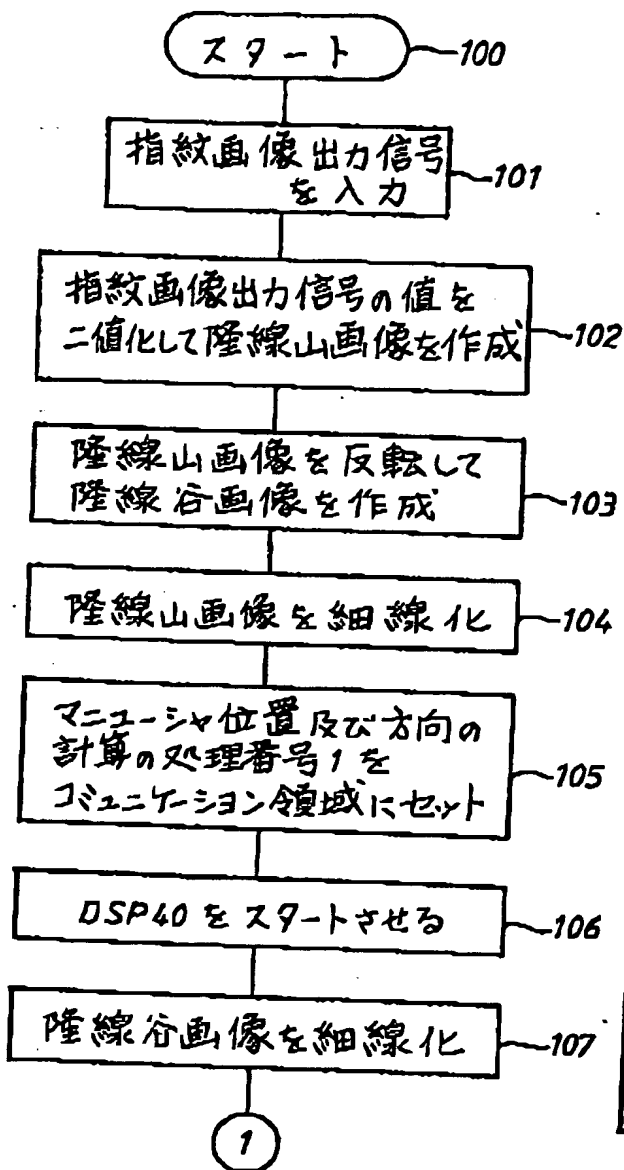




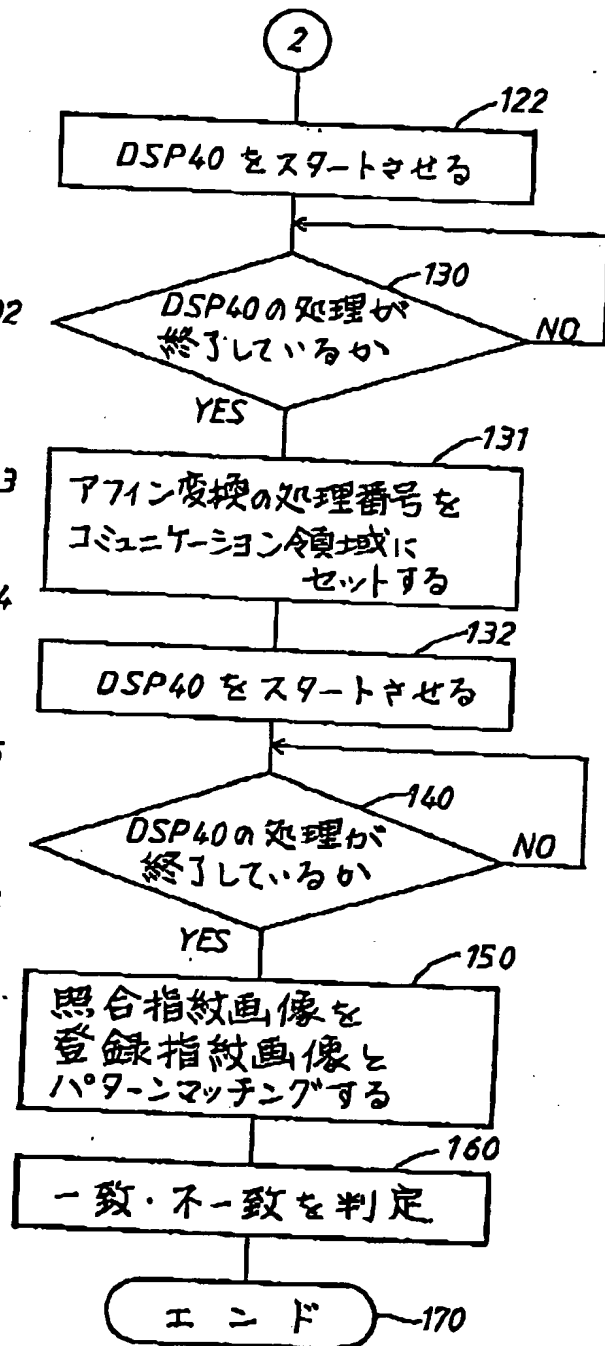
(9)

特開平5-81219

【図3】



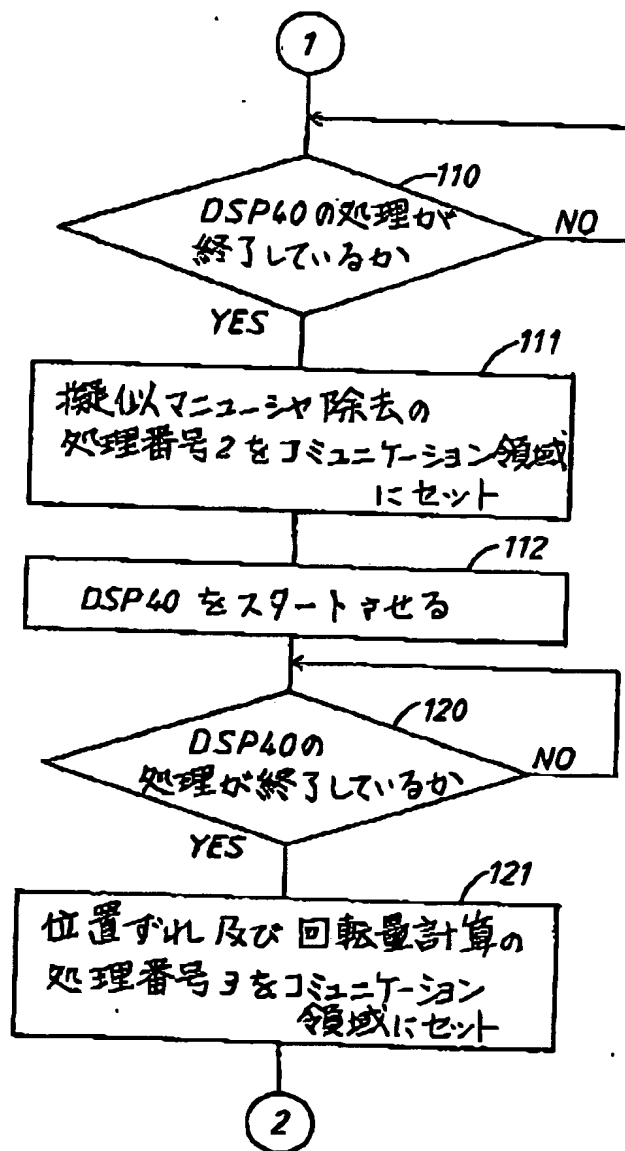
【図5】



(10)

特開平5-81219

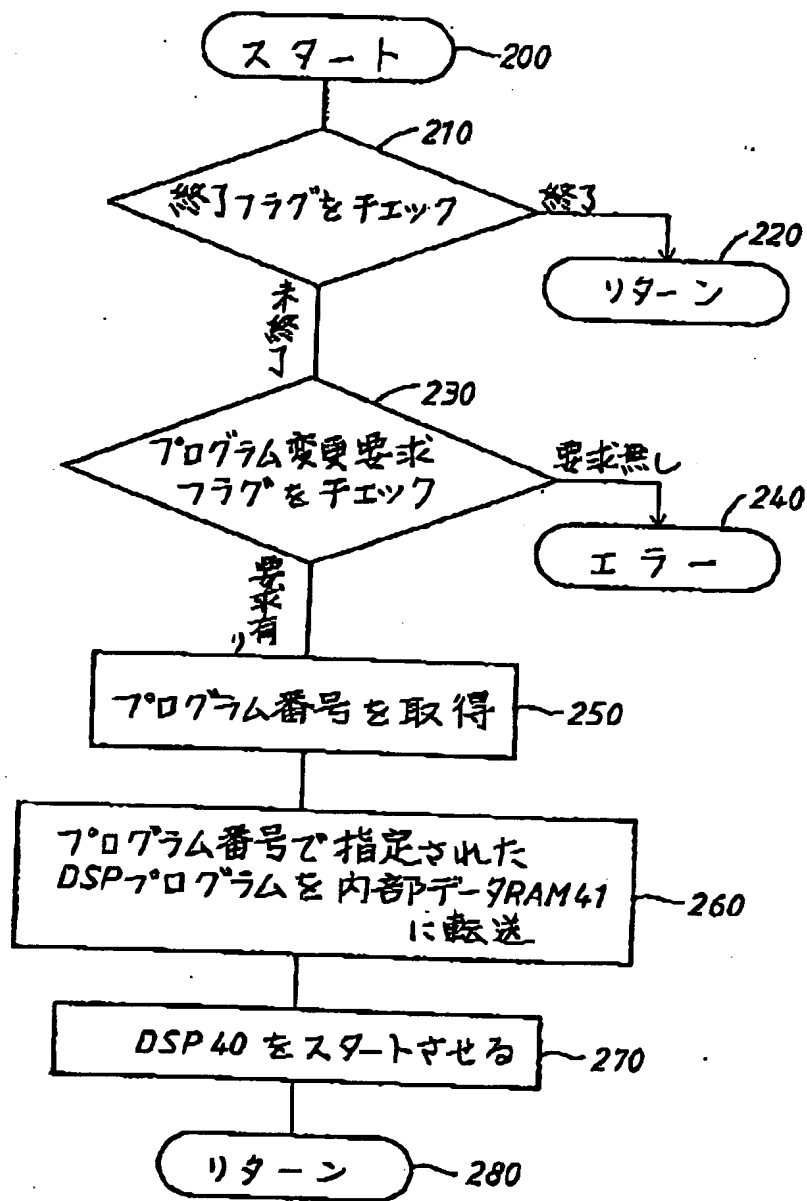
【図4】



(11)

特開平5-81219

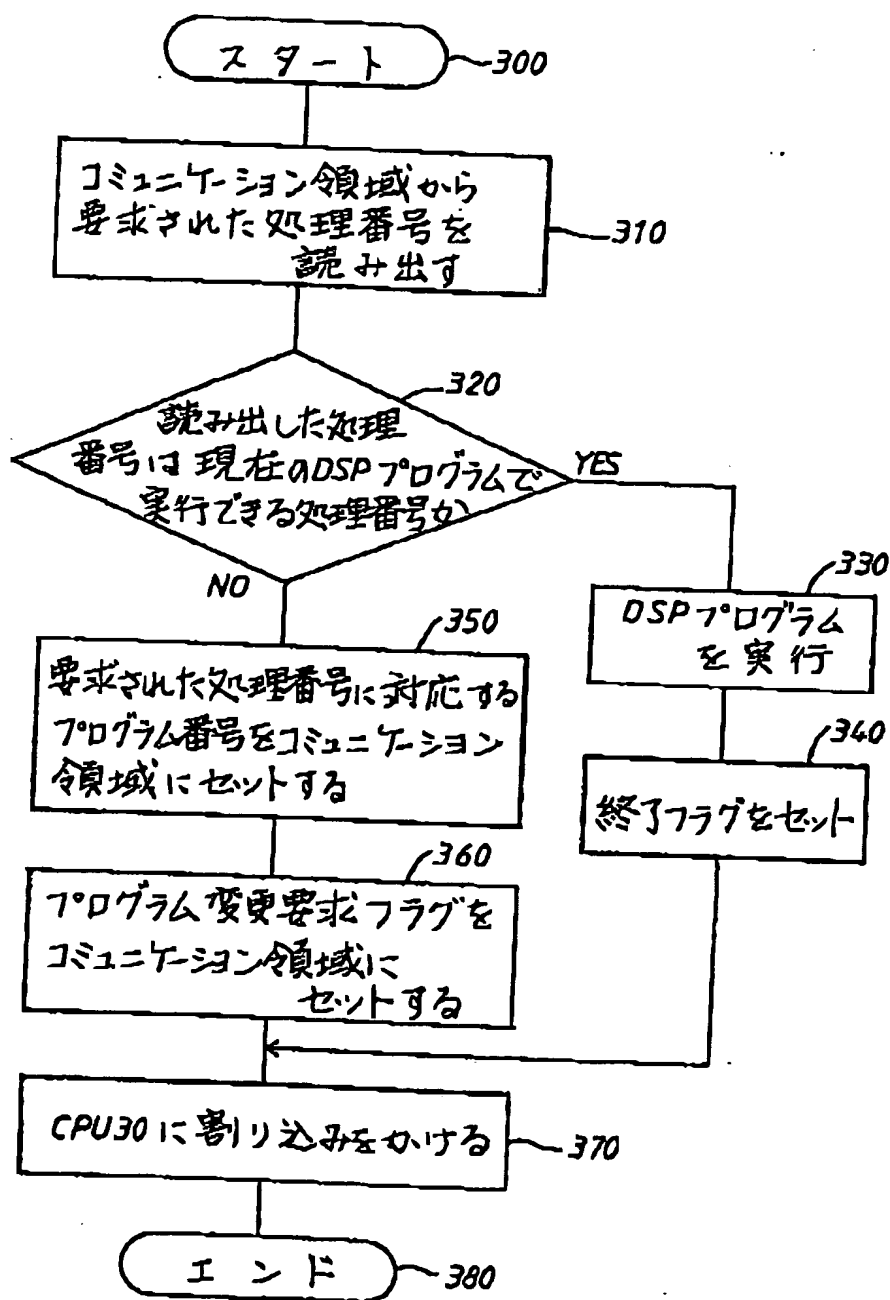
【図8】



(12)

特開平5-81219

【図7】



【手続補正書】

【提出日】平成3年11月22日

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】0011

【補正方法】変更

【補正内容】

【0011】次の表3は、DSP40のコミュニケーション領域にセットされる各データを表すもので、これら各データは、DSP40とCPU30との間で受け渡される。

(13)

特開平5-81219

【表3】

\* \*

処理番号
終了フラグ
プログラム変更要求フラグ
プログラム番号

(19) Japanese Patent Office (JP)  
(12) Publication of Laid Open Patent Application (A)  
(11) Publication Number of Patent Application:  
Japanese Patent Laid-Open Publication No. Hei 5  
(1993)-81219

(43) Date of Publication of Application  
April 2, 1993

(51) Int. Cl.<sup>5</sup>: G06F 15/16  
15/78

Identification Mark Intraoffice Reference Number: FI

420 S 9190-5L

510 A 7530-5L

Request for Substantive Examination:

not requested, Number of Claims: 1 (Total 13 pages)

(21) Application Number:

Japanese Patent Application No. Hei 3(1991)-273228

(22) Date of Filing: September 24, 1991

(71) Applicant: 000004260

NIPPONDNESO CO., LTD.

1, Showamachi 1-chome, Kariya-shi, Aichi

(72) Inventor: Takao Suzuki

c/o NIPPONDNESO CO., LTD.

1, Showamachi 1-chome, Kariya-shi, Aichi

Inventor: Toshitaka Kamiya

c/o NIPPONDNESO CO., LTD.

1, Showamachi 1-chome, Kariya-shi, Aichi

Inventor: Koji Kawasaki

c/o NIPPONDNESO CO., LTD.

1, Showamachi 1-chome, Kariya-shi, Aichi

(74) Agent: patent attorney Shoichi Hase

[Title of the Invention]

# APPARATUS FOR CONTROLLING OPERATION OF MICROPROCESSOR

[Abstract]

[Object]

In order to minimize a degree of lowering in an execution speed of a DSP due to a transmission of an instruction code from a host microprocessor to an internal instruction RAM of the DSP, the transmission of an instruction code is not implemented when the processing immediately previously implemented is to be executed once again.

[Constitution]

A CPU 30 selectively sets, in an internal data RAM 41, one of processing-numbers respectively assigned to processing contents executed by a DSP 40. Each time one of processing-numbers is thus set, an operation of the DSP 40 is initiated, and each time the operation of the DSP 40 is thus initiated, the DSP 40 determines whether the processing-number set in the internal data RAM 41 corresponds to a program-number assigned to a DSP program which is stored in an internal instruction RAM 42. Where the processing-number and the program-number correspond, the DSP 40 executes the processing content according to the DSP program in the internal instruction RAM 42. On the other hand, where the processing-number and the program-number do not correspond, a host microprocessor transmits a DSP program of a program-number corresponding to the processing-number which has been set in the internal data



RAM 41, to the internal instruction RAM 42, to store the DSP program therein.

[Claims]

[Claim 1]

An apparatus for controlling an operation of a microprocessor, comprising: a host microprocessor; and a DSP which comprises an internal data RAM and an internal instruction RAM, and serves as a slave microprocessor,

wherein the host microprocessor comprises:

a memory in which are stored a plurality of DSP programs according to each of which at least one corresponding processing content of the DSP is executed, together with program-numbers assigned to the respective processing contents;

processing-number setting means which selectively sets one of processing-numbers respectively assigned to the processing contents of the DSP, in the internal data RAM; and

starting means which makes the DSP to start operating, after the processing-number setting means sets the one of the processing-numbers,

wherein the DSP comprises:

determining means which determines, upon the start of the operation of the DSP, whether the processing-number set in the internal data RAM corresponds to a program-number assigned to one of the DSP programs which is stored in the internal instruction RAM; and

executing means which executes, when the determining means determines that the processing-number set in the internal data RAM corresponds to the program-number of the DSP program stored in the internal instruction RAM, the processing content of the said processing-number, according to the DSP program stored in the internal instruction RAM,

and wherein when the determining means determines that the processing-number set in the internal data RAM does not correspond to the program-number of the DSP program stored in the internal instruction RAM, the host microprocessor transmits one of the DSP programs of the program-numbers which corresponds to the processing-number set in the internal data RAM, from the memory to the internal instruction RAM.

[Detailed Description of the Invention]

[0001]

[Industrial Field of the Invention]

The present invention relates to a microprocessor comprising an instruction random access memory (hereinafter referred to as "internal instruction RAM") in a chip, and operates in accordance with instructions read from the internal instruction RAM. In particular, the invention relates to an apparatus for controlling an operation of a microprocessor, which is suitable for controlling instruction execution by a digital signal processor (hereinafter referred to as "DSP") included in the microprocessor.

[0002]

[Prior Art]

In a conventional DSP comprising an internal instruction RAM and functioning as a slave microprocessor, it is typical that a central processing unit (hereinafter referred to as CPU) of a host microprocessor reads an instruction code from a memory incorporated in the host microcomputer and transmits it to the internal instruction RAM, prior to the execution of the instruction.

[0003]

[Problem to be Solved by the Invention]

In such a DSP, it is typical that the capacity of the internal instruction RAM is limited to a scale of several hundreds of steps, due to limitation in the chip area. Accordingly, in a case where a large number of processing contents is executed by a single DSP, an inconvenience that changeover of the instruction code in the internal instruction RAM is required, is encountered. However, the arrangement where the instruction code stored in the internal instruction RAM is changed over prior to every execution of processing content by the DSP, invites undesirable lowering in an execution speed of the DSP. Further, when a change is made to instruction codes or properties of programs of the DSP in the development phase, the CPU has to know the change so as to transmit the instruction codes. Thus, changing the programs of the DSP inevitably involves changing the programs of the CPU. In order to deal with the above-described drawback, an object of the invention is to provide an apparatus for controlling an

operation of a microprocessor, wherein the lowering in the execution speed of a DSP due to a transmission of an instruction code from a host microprocessor to an internal instruction RAM of the DSP is minimized, by making it unnecessary to transmit the instruction code where the processing as previously executed is to be executed again, and by making it unnecessary to make a change to a program of the host microprocessor even when a change is made to a program of the DSP.

[0004]

[Means for Solving the Problem]

To solve the above problem, the invention provides an apparatus for controlling an operation of a microprocessor, the apparatus comprising a host microprocessor, and a DSP which comprises an internal data RAM and an internal instruction RAM, and serves as a slave microprocessor, wherein the host microprocessor comprises: a memory 1 in which are stored a plurality of DSP programs according to each of which at least one corresponding processing content of the DSP is executed, together with program-numbers assigned to the respective processing contents; processing-number setting means 2 which selectively sets one of processing-numbers respectively assigned to the processing contents of the DSP, in the internal data RAM; and starting means 3 which makes the DSP to start operating, after the processing-number setting means 2 sets the one of the processing-numbers, wherein the DSP comprises: determining means 4 which determines, upon the start of the operation of the DSP, whether the processing-number set in the

internal data RAM corresponds to a program-number assigned to one of the DSP programs which is stored in the internal instruction RAM; and executing means 5 which executes, when the determining means 4 determines that the processing-number set in the internal data RAM corresponds to the program-number of the DSP program stored in the internal instruction RAM, the processing content of the said processing-number, according to the DSP program stored in the internal instruction RAM, and wherein when the determining means 4 determines that the processing-number set in the internal data RAM does not correspond to the program-number of the DSP program stored in the internal instruction RAM, the host microprocessor transmits one of the DSP programs of the program-numbers which corresponds to the processing-number set in the internal data RAM, from the memory 1 to the internal instruction RAM.

[0005]

[Operation]

Each time the processing-number setting means 2 of the host microprocessor selectively sets, in the internal data RAM, one of the processing-numbers respectively assigned to the plurality of processing contents of the DSP, the starting means makes the DSP to start operating. Each time the operation of the DSP is thus started, the determining means 4 of the DSP determines whether the processing-number set in the internal data RAM corresponds to the program-number of the DSP program stored in the internal RAM. Where the determining member determines that the processing-number and program-number

correspond, the executing means 5 executes the processing content according to the DSP program in the internal instruction RAM. On the other hand, where the determining means 4 determines that they do not correspond, the host microprocessor transmits from the memory 1 the DSP program of the program-number corresponding to the processing-number set in the internal data RAM, to store the DSP program therein. Then, based on the determination by the determining means 4 that the processing- and program-numbers correspond, the executing means 5 executes the processing content according to the DSP program transmitted to and stored in the internal instruction RAM.

[0006]

[Effects of the Invention]

As described above, unless the determining means 4 determines that the processing- and program-numbers do not correspond, the DSP program stored in the internal instruction RAM is not changed over, but is used in the processing executed by the executing means 5. Hence, without increasing the capacity of the internal instruction RAM, the lowering in the execution speed of the DSP, due to the changeover of the DSP program in the internal instruction RAM, is minimized. Further, the changeover of the DSP program in the internal instruction RAM is implemented simply such that the host microprocessor transmits the DSP program of interest from the memory 1 to the internal instruction RAM, based on the determination by the determining means 4 that the processing- and program-numbers

do not correspond. Therefore, it is made unnecessary to change the program in the host microprocessor.

[0007]

[Embodiments]

There will be described one embodiment of the invention by reference to the drawings. Fig. 2 shows an example where the present invention is applied to a fingerprint matching system. This fingerprint matching system includes an image input device 10 having a read screen as an image sensor. When a tip of a finger of an object person whose fingerprint is to be matched is pressed onto the read screen, the fingerprint is optically read, and a fingerprint-image output signal is generated based on the read image of the fingerprint. An interface 20 operates to output the fingerprint-image output signal received from the image input device 10 to a CPU 30 of a host microprocessor. The CPU 30 cooperates with the interface 20, a DSP 40, a DMA controller 60 and etc. to execute a main control program and an interrupt control program according to flowcharts shown in Figs. 3-6, during which arithmetic processing requisite for fingerprint matching is performed. The CPU 30 incorporates a memory (hereinafter referred to as "CPU memory") whose memory area 33 (hereinafter referred to as "CPU memory area 33") includes a DSP instruction code holding area 33a where instruction codes for the DSP 40 are stored, namely, four DSP programs 1, 2, 3 and 4. In addition, in the CPU memory area 33 are stored the main control program and interrupt control program.

[0008]

The DSP 40 serves as a slave microprocessor, and incorporates a data RAM 41 (hereinafter referred to as "internal data RAM") having a communication area, and an internal instruction RAM 42 for temporarily storing a content of an instruction issued to the DSP 40. The DSP 40 executes a DSP control program according to a flowchart shown in Fig. 7, during which various kinds of arithmetic processing are performed. Upon termination of each kind of arithmetic processing, an interrupt to the CPU 30 is generated, by generating an interrupt signal which is outputted to the CPU 30. The DSP 40 outputs data to an external data RAM 45 via a DSP address bus 43 and a DSP data bus 44 to temporarily store the data in the external data RAM 45. The DSP control program is stored in the DSP 40.

[0009]

Numbers (processing-numbers 1-4) assigned to respective kinds of processing (which will be referred to as "processing contents") and numbers (program-numbers 1-3) assigned to respective DSP programs correspond, in relation to the processing contents executed by the DSP 40, as shown in Table 1 below:



[Table 1]

Processing -number	Processing Content	Program -number
1	Calculate Position and Orientation of Minutiae	1
2	Eliminate False Minutiae	2
3	Calculate Position Error and Amount of Rotation	2
4	Affine Transformation	3

Note that in Table 1 the processing-numbers 1-4 correspond to the processing contents of the DSP 40 in one-to-one relationship, but it is not the case with the processing contents of the DSP 40 and the program-numbers. Two processing contents "Eliminate False Minutiae" and "Calculate Position Error and Amount of Rotation" correspond to a single program number, namely, program-number 2. Program-number 1-3 represents that the respectively corresponding processing content of the DSP 40 is contained in a program of that number as stored in the DSP instruction code holding area 33a.

[0010]

Table 2 below is a program-number table where the program-numbers in Table 1 are listed in order of the corresponding processing-numbers, top to bottom. This program-number table is stored in the internal data RAM 41 of

the DSP 40.

[Table 2]

Offset Address	Content of Table
1	1
2	2
3	2
4	3

[0011]

Table 3 below shows data which are set in the communication area of the DSP 40 and transferred between the DSP 40 and the CPU 30.

[Table 3]

Processing-number
Ending flag
Program change request flag
Program-number

[0012]

When the CPU 30 needs to access the DSP 40, an address decoder 50 decodes an access instruction from the CPU 30 into a DSP selecting signal, which is outputted to the DSP 40. Further, when the CPU 30 accesses a direct memory access

controller 60 (hereinafter referred to as "DMA controller 60"), the address decoder 50 decodes an access instruction from the CPU 30 into a DMA selecting signal which is outputted to the DMA controller 60. In response to the DMA selecting signal from the address decoder 50 and a DMA signal from the CPU 30, the DMA controller 60 transmits required data from the CPU 30 to the internal data RAM 41 and the internal instruction RAM 42 through a CPU data bus 31.

[0013]

According to the above-described embodiment, when the system of the invention is placed in its operated state, the CPU 30 initiates execution of the main control program according to the flowchart of Figs. 3-5, beginning with step 100. When a fingerprint-image output signal is presently outputted to the interface 20, the fingerprint-image output signal is inputted from the interface 20 to the CPU 30 in step 101. In the following step 102, the CPU 30 binarizes the fingerprint-image output signal to generate a positive image of a ridge pattern of the fingerprint in question, and in step 103 the positive image of the ridge pattern is inverted to generate a negative image of the ridge pattern of the fingerprint. Then in step 104, the positive image of the ridge pattern generated in step 102 is subjected to a thinning operation, and in step 105, processing-number 1 corresponding to one of the processing contents of the DSP 40, i.e., calculation of the position and orientation of minutiae, is set in the communication area of the internal data RAM 41 of the

DSP 40, to proceed to the next step 106 of the main control program.

[0014]

When the main control program thus proceeds to step 106, the CPU 30 initiates operation of the DSP 40, and in the next step 107 the negative image of the ridge pattern obtained in step 103 is subjected to a thinning operation. Then, a negative decision "NO" is repeatedly made for a determination in step 110. The initiating operation of the DSP 40 as mentioned above is such that the DSP 40 initiates execution of the DSP control program according to the flowchart in Fig. 7, beginning with step 300. Then, in the next step 310, the DSP 40 reads processing-number 1 as has been previously set in the communication area of the internal data RAM 41, and the DSP control program proceeds to step 302.

[0015]

On the other hand, when processing-number 1 as read out in step 310 does not coincide with the number of processing content executable according to the DSP program which has been transmitted to the internal instruction RAM 42, the DSP 40 makes a negative decision "NO" in step 320, and, in step 350, references the program-number table in the internal data RAM 41 so that program-number 1 corresponding to processing-number 1 which has been read out in step 310 is set in the communication area of the internal data RAM 41. Then, a program change request flag is set in the communication area in step 360, and the DSP control program proceeds to step 370.

[0016]

When the DSP control program thus proceeds to step 370, the DSP 40 generates an interrupt to the CPU 30, which then suspends the execution of the main control program and initiates execution of an interrupt control program according to the flowchart of Fig. 6, beginning with step 200. In step 210, whether the ending flag is set or not is checked. Since the ending flag is not presently set, with the negative decision "NO" made in step 320 as described above, the CPU 30 determines that the ending flag is not set. With this decision in step 210 that the DSP program is not ended, it is checked whether the program change request flag is set or not, in step 230. Since the program change request flag has been set in step 360 as described above, the CPU 30 determines that a changeover of the program is requested, and obtains program-number 1 from the communication area of the internal data RAM 41, in step 250.

[0017]

In step 260, to transmit the DSP program as designated by program-number 1 to the internal instruction RAM 42 of the DSP 40, the CPU 30 makes the address decoder 50 to output the DMAC selecting signal to the DMA controller 60, which reads out DSP program 1 from the DSP instruction code holding area 33a of the CPU memory area 33 of the CPU 30, and transmits the program to the internal instruction RAM 42 where the program is newly stored. Then the CPU 30 makes the DSP 40 to start operating in step 270, and in the next step 280 the execution of the interrupt control program is terminated.

[0018]

When the operation of the DSP 40 is initiated in step 270 as described above, the DSP 40 initiates execution of the DSP control program according to the flowchart of Fig. 7, beginning with step 300, and processing-number 1 in the communication area of the internal data RAM 41 is read out, in step 310. Since the read processing-number 1 corresponds to the DSP program which has been transmitted to the internal instruction RAM 42 in step 260, the DSP 40 makes an affirmative decision "YES" in step 320. In step 330, based on DSP program 1 which has been transmitted to the internal instruction RAM 42, the DSP 40 executes processing of calculating the position and orientation of the minutiae based on the results of the thinning operation performed on the positive image of the ridge pattern. Then in step 340, the DSP 40 sets the ending flag in the communication area of the internal data RAM 41, and generates an interrupt to the CPU 30 in step 370.

[0019]

Accordingly, the CPU 30 initiates execution of the interrupt control program in the same way as described above, and in step 210 it is determined that the ending flag is set, since the ending flag has been set in step 340, and the execution of the interrupt control program is terminated in step 220. Thereafter, the CPU 30 makes an affirmative decision "YES" in step 110 of the main control program, based on that the ending flag has been set in step 340, and sets processing-number 2 corresponding to elimination of false minutiae, in the

communication area of the internal data RAM 41 in step 111. The CPU 30 then makes the DSP 40 to start operating in the next step 112, and repeatedly makes a negative decision "NO" in step 120. [0020]

When the DSP 40 starts operating in step 112 as described above, the DSP 40 reads processing-number 2 as has been set in the communication area of the internal data RAM 41, in step 310 of the DSP control program according to the flowchart of Fig. 7. The DSP control program then proceeds to step 320. Since at this time processing-number 2 which has been read in step 310 does not correspond to processing-number 1 executable according to the DSP program which has been transmitted to the internal instruction RAM 42, the DSP 40 makes a negative decision "NO" in step 320, and references the program-number table in the internal data RAM 41 in step 350. Then in step 310, the DSP 40 sets program-number 2 corresponding to processing-number 2 as has been read out in step 310, in the communication area of the internal data RAM 41, and sets the program change request flag in the communication area in step 360. The DSP control program now proceeds to step 370 to generate an interrupt to the CPU 30.

[0021]

When the interrupt to the CPU 30 is thus generated, the CPU suspends the execution of the main control program and goes on to execute the interrupt control program according to the flowchart of Fig. 6, and a negative decision "NO" is made in the determination of step 210, based on the negative decision

made in step 320 as described above. Then in step 230, the CPU determines that a changeover of the program is requested, since the program change request flag has been set in step 360 as described above. In step 250, the CPU obtains program-number 2 from the communication area of the internal data RAM 41. In step 260, to transmit the DSP program designated by program-number 2 to the internal instruction RAM of the DSP 40, the CPU 30 makes the address decoder 50 to output the DMAC selecting signal to the DMA controller 60, which then reads DSP program 2 from the DSP instruction code holding area 33a of the CPU memory area 33 of the CPU 30, and transmits the program to the internal instruction RAM 42 where the program is newly stored. Then, the CPU 30 makes the DSP 40 to start operating in step 270, and terminates the execution of the interrupt control program in the next step 280.

[0022]

When the DSP 40 starts operating in step 270 as described above, the DSP 40 executes the DSP control program according to the flowchart of Fig. 7, and processing-number 2 in the communication area of the internal data RAM 41 is read out. Since at this time processing-number 2 read out as described above corresponds to the DSP program as has been transmitted to the internal instruction RAM 42 in step 260, the DSP 40 makes an affirmative decision "YES" in step 320, and executes, in step 330, the processing of eliminating the false minutiae based on the results of the calculation of the position and orientation of the minutiae as described above, according



to DSP program 2 which has been transmitted to the internal instruction RAM 42. In step 340, the DSP 40 sets the ending flag in the communication area of the internal data RAM 41, and generates an interrupt to the CPU in step 370. Accordingly, the CPU 30 initiates execution of the interrupt control program in the same way as described above, and determines that the execution of the DSP program is terminated, based on that the ending flag has been set in step 340, and then terminates the execution of the interrupt control program in step 220.

[0023]

Thereafter, the CPU 30 makes an affirmative decision in step 120 of the main control program, based on that the ending flag has been set in step 340, and sets processing-number 3 corresponding to calculation of position error and amount of rotation of the minutiae, in the communication area of the internal data RAM 41, in step 121. The CPU 30 makes the DSP 40 to start operating in the next step 122, and repeatedly makes a negative decision "NO" in step 130. When the operation of the DSP 40 is thus started in step 122, the DSP 40 reads processing-number 3 as has been set in the communication area of the internal data RAM 41, in step 310 of the DSP control program according to the flowchart of Fig. 7, and the DSP control program proceeds to step 320.

[0024]

Since at this time processing-number 3 which is read out as described above corresponds to the DSP program as has been transmitted to the internal instruction RAM 42 in step 260,

the DSP 40 makes an affirmative decision "YES" in step 320, and executes, in step 330, the processing of calculating the position error and amount of rotation, based on the results of the calculation of the position and orientation of the minutiae as described above, and in accordance with the DSP program 2 as has been transmitted to the internal instruction RAM 42. Then in step 340, the DSP 40 sets the ending flag in the communication area of the internal data RAM 41, and generates an interrupt to the CPU 30 in step 370. Accordingly, the CPU 30 initiates execution of the interrupt control program in the same way as described above, and determines that the DSP program is ended in step 210, since the ending flag has been set in step 340, and then terminates the execution of the interrupt control program in step 220.

[0025]

Thereafter, the CPU 30 determines in step 340 of the main control program that the processing by the DSP 40 is terminated, based on that the ending flag has been set in step 340, and sets processing-number 4 corresponding to affine transformation, in the communication area of the internal data RAM 41, in step 131. In the next step 132, the CPU 30 makes the DSP 40 to start operating, and repeatedly makes a negative decision "NO" in step 140. When the operation of the DSP 40 is thus started in step 132, the DSP 40 reads processing-number 4 as has been set in the communication area of the internal data RAM 41, in step 310 of the DSP control program according to the flowchart of Fig. 7. Thus, the DSP control program proceeds

to step 320.

[0026]

Since at this time processing-number 4 read out as described above does not correspond to the DSP program as has been transmitted to the internal instruction RAM 42 in step 260, the DSP 40 makes a negative decision "NO" in step 320, and references the program-number table in the internal data RAM 41 and sets program-number 3 corresponding to processing-number 4 as has been read out in step 310, in the communication area of the internal data RAM 41, in step 350. Then in step 360, the DSP 40 sets the program change request flag in the communication area, and the DSP control program proceeds to step 370 to generate an interrupt to the CPU 30.

[0027]

Accordingly, the CPU 30 suspends the execution of the main control program, and executes the interrupt control program according to the flowchart of Fig. 6. In step 210, the CPU 30 determines that the DSP program is not ended, in the same way as described above, and it is determined, in step 230, that a changeover of the program is requested, since the program change request flag has been set in step 360 as described above. In step 250, the CPU 30 obtains program-number 3 from the communication area of the internal data RAM 41, in step 250. In step 260, to transmit the DSP program designated by program-number 3 to the internal instruction RAM 42 of the DSP 40, the CPU 30 makes the address decoder 50 to output the DMAC selecting signal to the DMA controller 60, which reads DSP

program 3 from the DSP instruction code holding area 33a of the CPU memory area 33 of the CPU 30, and transmits the program to the internal instruction RAM 42 where the program is newly stored. Then, the CPU 30 makes the DSP 40 to start operating in step 270, and terminates the execution of the interrupt control program in the next step 280.

[0028]

When the operation of the DSP 40 is thus started in step 270, the DSP 40 executes the DSP control program according to the flowchart of Fig. 7, and reads processing-number 4 in the communication area of the internal data RAM 41 in step 310. Since at this time processing-number 4 read out as described above corresponds to the DSP program which has been transmitted to the internal instruction RAM 42 in step 260, the DSP 40 makes an affirmative decision "YES" in step 320, and executes, in step 330, the processing of the affine transformation, based on the results of the calculation of the position error and amount of rotation, and according to DSP program 3 as has been transmitted to the internal instruction RAM 42. In step 340, the DSP 40 sets the ending flag in the communication area of the internal data RAM 41, and generates an interrupt to the CPU 30 in step 370.

[0029]

Accordingly, the CPU 30 initiates execution of the interrupt control program in the same way as described above, and determines that the DSP program is ended in step 210, since the ending flag has been set in step 340, and terminates the

execution of the interrupt control program in step 220. Thereafter, the CPU 30 makes an affirmative decision "YES" in step 140, in the substantially same way as described above, and performs pattern-matching between the registered fingerprint image of the object person and the fingerprint image after subjected to the affine transformation, in step 150. In step 160, the CPU determines whether the registered fingerprint image and the fingerprint image in question match, based on the result of the pattern-matching. The registered fingerprint image is previously stored in the CPU memory area of the CPU 30.

[0030]

As illustrated above, in the calculation processing cooperatively implemented by the CPU 30 and DSP 40 during the matching operation is performed on the fingerprint of the object person, the processing of eliminating the false minutiae and the processing of calculating the position error and amount of rotation are implemented in series, as described above. Through the calculation processing in steps 111-122 in the main control program as well as implementation of the DSP control program, each of the processing of eliminating the false minutiae and the processing of calculating the position error and amount of rotation are implemented while DSP program 2 in the internal instruction RAM 42 is maintained without being changed over. Hence, the number of times when the DSP program in the internal instruction RAM 42 is changed is minimized. Accordingly, the capacity of the internal instruction RAM 42

can be minimized while the execution speed of the DSP 40 is improved. In the present arrangement, the changeover of the DSP program in the internal instruction RAM does not require making a change to a program in the CPU 30. Further, the calculation of the position and orientation of the minutiae in the positive image of the ridge pattern is implemented in step 330, by the DSP 40 which has started operating in step 106 after the thinning operation is performed on the positive image of the ridge pattern in step 104 of the main control program, concurrently with that the CPU 30 executes the thinning operation on the negative image of the ridge pattern. Therefore, the overall execution speed can be enhanced by the cooperation of the CPU 30 and DSP 40.

[0031]

It is noted that where the present invention is embodied without employing the DMA controller 60, access to the DSP 40 from the CPU 30 can be made by I/O read/write. Further, the application of the invention is not limited to the fingerprint matching system, but may be applied to any other system comprising a DSP and a host CPU.

[Brief Description of the Drawings]

[Fig. 1] Fig. 1 is a diagram corresponding to the description of claims.

[Fig. 2] Fig. 2 is a block diagram of a fingerprint matching system where the invention is applied.

[Fig. 3] Fig. 3 shows an initial part of a flowchart of a main control program executed by a CPU shown in Fig. 2.

[Fig. 4] Fig. 4 shows an intermediate part of the flowchart of the main control program.

[Fig. 5] Fig. 5 shows a final part of the flowchart of the main control program.

[Fig. 6] Fig. 6 shows a flowchart of an interrupt control program executed by the CPU of Fig. 2.

[Fig. 7] Fig. 7 shows a flowchart of a DSP control program executed by a DSP of Fig. 2.

[Description of Reference Numerals]

- 30 CPU
- 33 CPU memory area
- 40 DSP
- 41 internal data RAM
- 42 internal instruction RAM

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**